Partitioning Large Requests Drill-Down

Mark A. Huber

mhuber@eos.hitc.com

2 November 1995

Overview

Request Partitioning

- Builds on Release A Approach
- Required to Control and Balance Large Requests
- Release B is Automated Approach
- Require DAAC Involvement on Issue

ECS Context

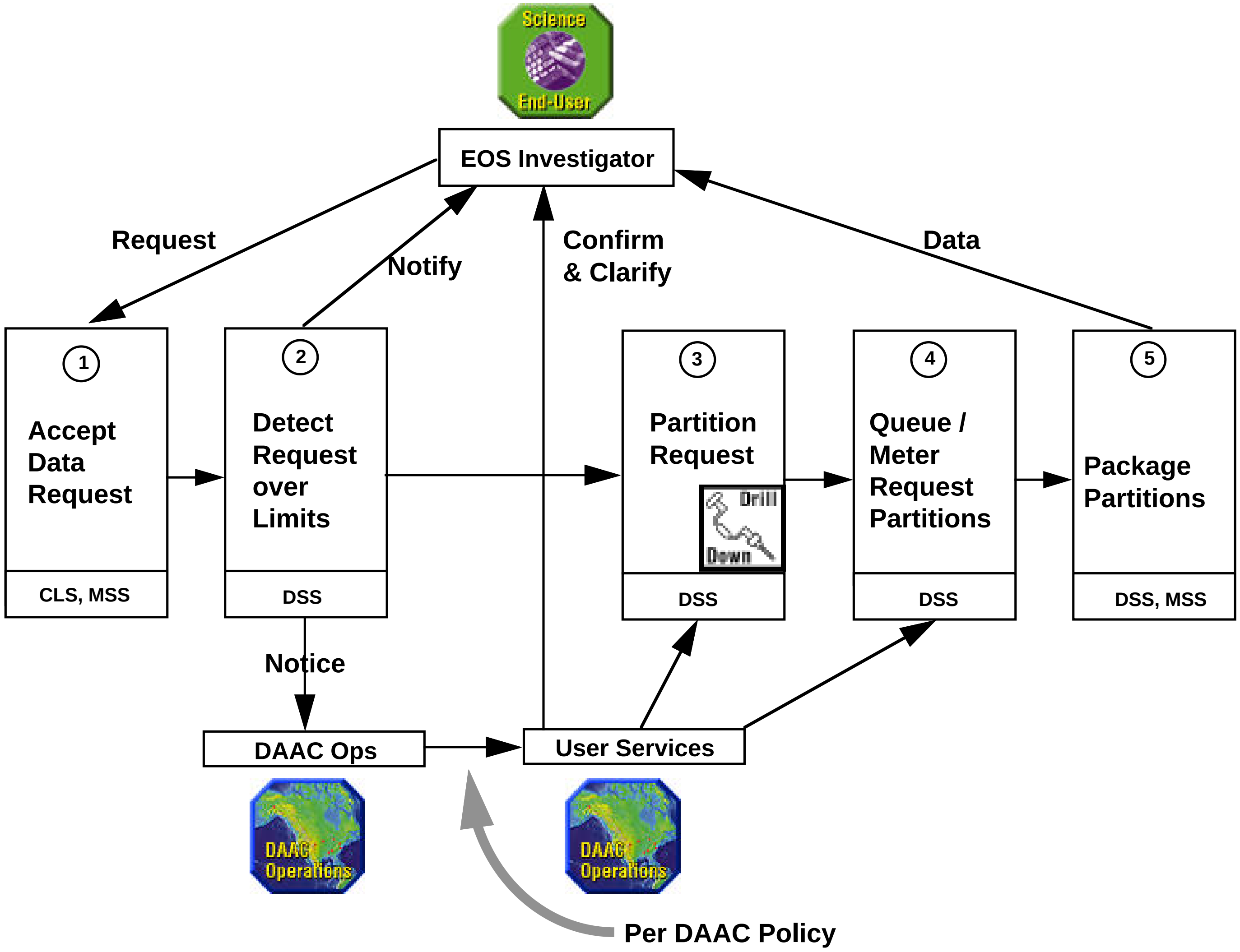
- Data Server

Scenario Context

- Push + Pull
  - Large Data Request

Huge Data Request

Functional Flow



Design Drivers

Architectural Drivers

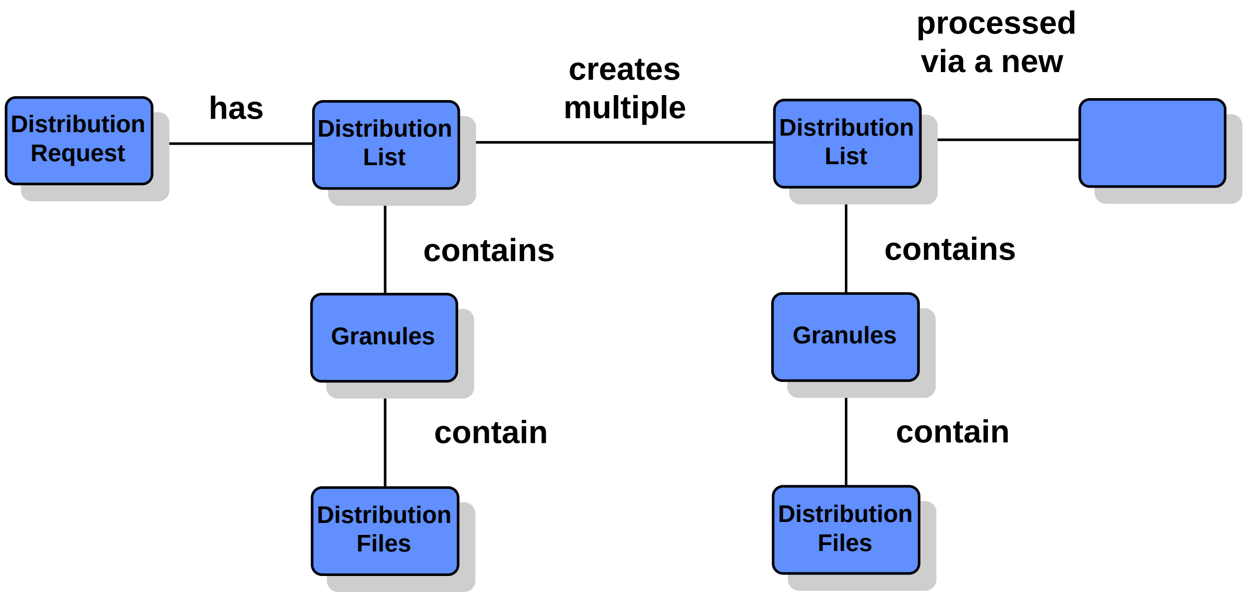
- Large Requests Happen (Historical)
- Operator Control/Management of System
  - Balance Resource Needs of Requests
  - Support Requests that Exceed Resources

Approach

- Request Flagged for Total Volume and/or Number of Files
- Release A is Manual
- Release B is Defining Degree of Automation

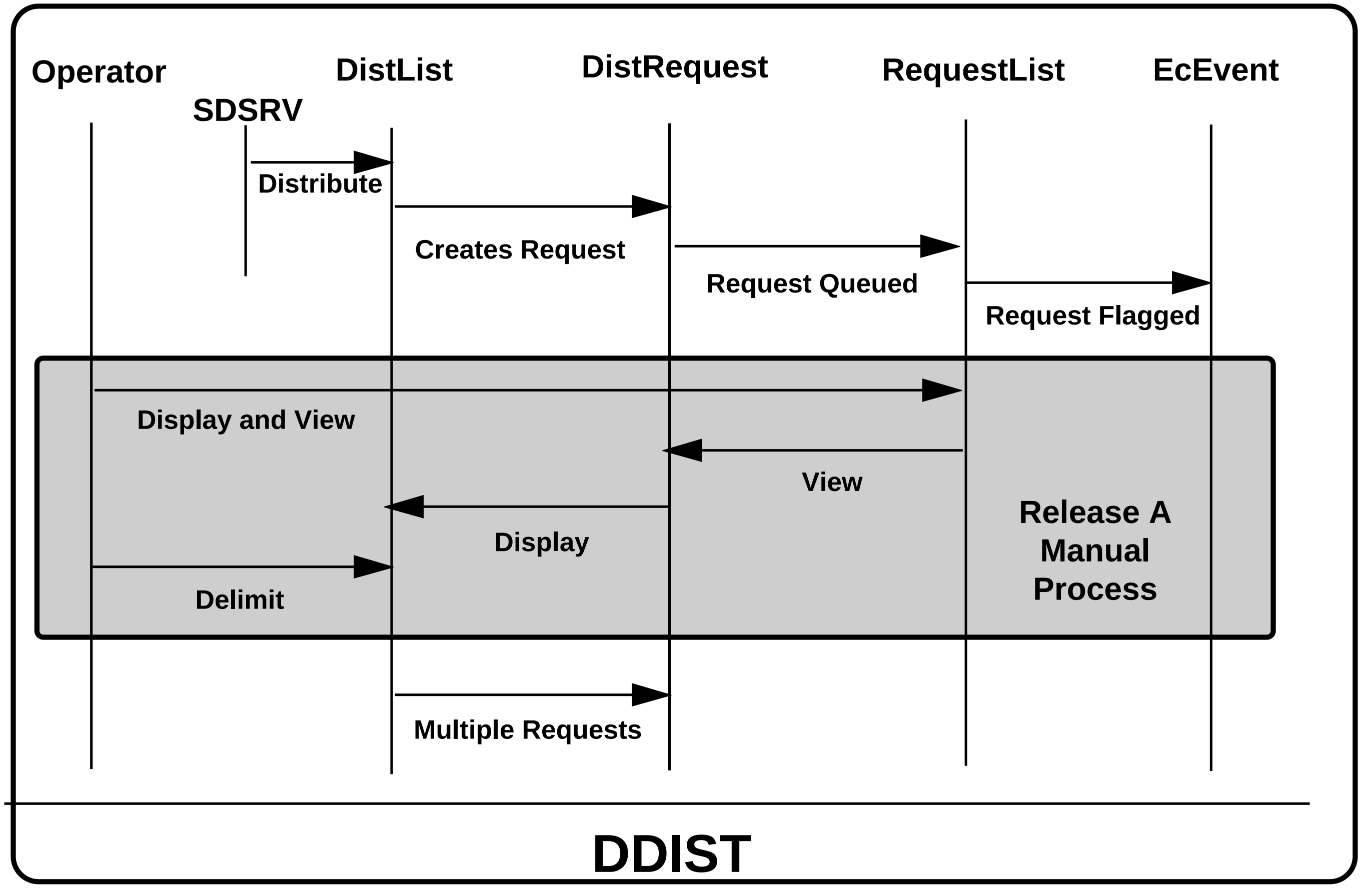
Software Design

High Level Class Model



For more details reference DID 305, Volume 24 (305-CD-024-001)

Software Design
High Level Event Trace



Evolutionary Features

Scalability

- Partitioning Rules can be Augmented as Conditions Change
- Approach Allows for Scaling of Requests without Regard to Physical System Constraints

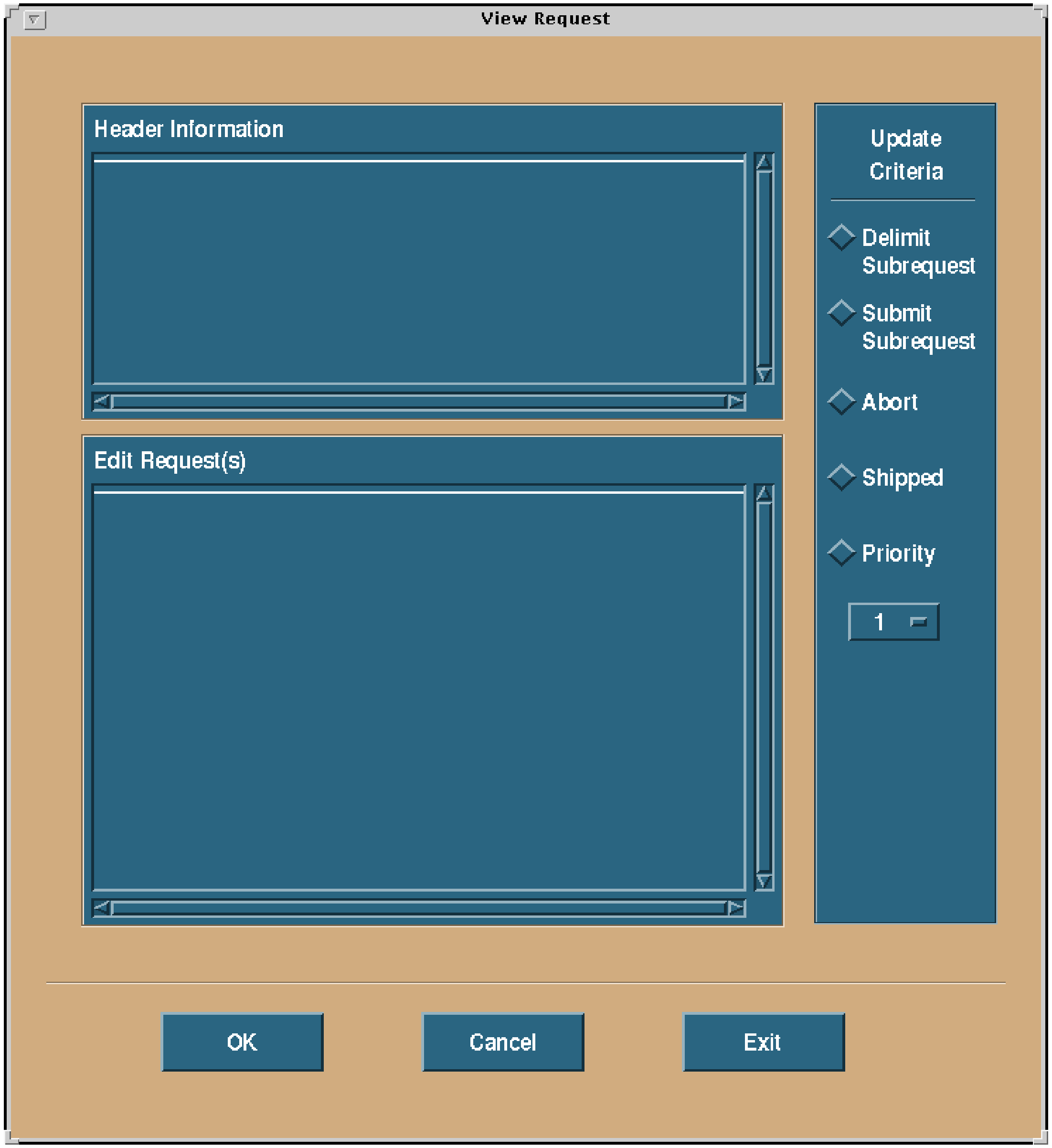
Current Status

Next Steps

- Request Mgmt/Partitioning Model is being Built for Release A
  - Lessons Learned Will Feed Release B

- Discuss Automated Approach via Telecons
  - Need DAAC Feedback on Approaches
  - DAAC Feedback will be sought at Ops Workshop

Preliminary Release A Ops Screen



Summary

Large Request Partitioning Highlights

- Builds on Release A Work
- Release B to Provide Automation Framework
- Will Work with DAACs to Define Candidate Algorithm(s)